University of Salford MANCHESTER

Pruning classification rules with instance reduction methods

Othman, O and Bryant, CH

http://dx.doi.org/10.7763/IJMLC.2015.V5.505

Title	Pruning classification rules with instance reduction methods
Authors	Othman, O and Bryant, CH
Type	Article
URL	This version is available at: http://usir.salford.ac.uk/33262/
Published Date	2015

Pruning Classification Rules with Instance Reduction Methods

Osama M. Othman and Christopher H. Bryant

***Abstract*— Generating classification rules from data often leads to large sets of rules that need to be pruned. A new pre-pruning technique for rule induction is presented which applies instance reduction before rule induction. Training three rule classifiers on datasets that have been reduced earlier with instance reduction methods leads to a statistically significant lower number of generated rules, without adversely affecting the predictive performance. The search strategies used by the three algorithms vary in terms of both type (depth-first or beam search) and direction (general-to-specific or specific-to-general).**

***Index Terms*—Rule Induction, Noise Filtering, Instance Reduction.**

I. Introduction

Our work is concerned with reducing the complexity of the rule-set by reducing the number of generated rules without adversely affecting the predictive accuracy.

We will consider rule induction methods that learn a set of propositional rules where the target concept is represented as a set of "if ... then ..." rules. Each rule consists of an antecedent (or body of rule) and a consequent. The consequent represents the predicted class; the antecedent part is composed of a conjunction of conditions, each involving one attribute. We focus on rule induction methods which produce an unordered set of rules because we are interested in rule-sets where each rule can be understood independently.

Most rule based systems tend to induce quite a large number of rules, making the solution obtained difficult to understand. The aim of our work is to investigate whether the number of generated rules can be reduced by preceding rule induction with instance reduction. We focus on instance reduction methods which have proved capable of reducing the size of training set and resulted in the smallest reduction in predictive accuracy [1], [2]. More specifically, we will apply algorithms that try to remove the border instances, which tend to be noisy instances or hard-to-learn, untypical instances.

The paper is organized as follows. Section 2 gives a short description of typical methods for rule induction. Section 3 reviews the instance reduction techniques we use in this work. In Section 4, we discuss the results of applying instance reduction before rule induction using CN2, PRISM and RISE in terms of predictive accuracy and number of generated rules. Section 5 presents our conclusions.

O. M. Othman School of Computing, Science and Engineering Newton Building, the University of Salford, Greater Manchester,M5,4WT,England,UK. (e-mail:O.Othman@edu.salford.ac.uk).

C. H. Bryant School of Computing, Science and Engineering, Newton Building, the University of Salford, Greater Manchester,M5,4WT,England,UK. (e-mail:C.H.Bryant@salford.ac.uk).

II. Rule Induction

Mitchell introduced the Candidate-Elimination algorithm, which served as the basis to develop the rule induction method. The rule induction method is to establish a hypothesis rule space which is based on a given example set and then to refine (search through) the hypothesis rule space to find more general rules [3].

There are many rule induction algorithms. Among them are AQ [4], CN2 [5] [6] and RIPPER [7]. All these algorithms employ the same general method that was used for the Candidate-Elimination algorithm. On other hand there are rule induction methods inspired by ideas from other methods like RULES (RULE Extraction System) which is a family of simple inductive learning algorithm inspired by ideas from both AQ and CN2. The RULES family is different from the other algorithms in that it does not induce rules on a class-per-class basis but instead considers the class of the selected seed example as the target class [8]. It then attempts to induce rules that cover as many examples of the target class as possible using the rule evaluation function.

Another approach of learning is to combine two or more different paradigms of learning in a single algorithm. RISE (Rule Induction from Set of Examples) [9] tries to combine the best characteristics of rule induction and instance based learning [10] in a single algorithm.

Other rule induction methods apply pruning methods during rule generations [11]. Fürnkranz and Widmer proposed a novel learning algorithm called IREP (Incremental Reduced Error Pruning) [12].

Some rule induction methods try to solve drawbacks of other induction methods. The PRISM [13] algorithm was proposed as an improvement to the ID3 [14] algorithm changing its principal induction strategy. ID3 produces its output in the form of decision tree. In [13], Cendrowska argues that decision trees can be incomprehensible, difficult to maintain and complicates the provision of explanation.

Table I compares some important characteristics of the afore-mentioned rule induction methods. This will guide us on selecting the algorithm that will be used in our experiments with pre pruning process.

We think that pre pruning process can achieve good results with rule induction algorithms which do not use pre-pruning such as CN2(modified), RISE, PRISM, AQ family, RULEs family and IREP. Also we can choose methods that have different type of search and different direction of search.

Accordingly we choose to investigate pruning on CN2 (modified), PRISM and RISE as it they have different type of search and direction of search.

TABLE I: COMPARING RULE INDUCTION METHODS

Rule Induction Method	Type of pruning	Direction of search	Type of search
AQ family	Post pruning	Hybrid	Beam search
CN2 (modified)	During rule generation	General to specific	Beam search
RIPPER	Pre and post pruning integration.	General to specific	Depth first
IREP	During rule generation	General to specific	Depth first
RULEs family	Post pruning	General to specific	Beam search
RISE	No	Specific to general	Depth first
PRISM	No	General to specific	Depth first

III. INSTANCE REDUCTION METHODS

Instance pruning tries to prune the original training set to get a smaller subset of it. Searching for a subset S of instances to keep instead of the original training set T can proceed in variety of directions, including: incremental, decremental and batch [2].

Incremental methods begin with empty subset S, and add instances (from training set T) to subset S if it fulfills some criteria. Thus if new instances are made available later (after training is completed) they can continue to be added to S according to the same criteria. Incremental methods are sensitive to the order of presentation of the instances. Condensed Nearest Neighbor (CNN) [15] and Selective Nearest Neighbor (SNN) [16] are examples of Incremental methods. On the other hand, decremental methods begin with all the instances in the training set (i.e., T=S), and search for instances to remove; they are often computationally more expensive than incremental methods. Reduced Nearest Neighbor (RNN) [17] and Decremental Reduction Optimization Procedure (DROP1-5) [1] represent examples of decremental methods. Finally, batch methods, as decremental methods, begin with all instances in training set, but before they remove any, they find all of the instances that meet the removal criteria and then they remove them all at once [18]. Batch methods also suffer from increased time complexity compared with incremental methods. In our experiments, we will use decremental and batch methods because, in comparison to incremental methods, they have been shown to give rise to higher predictive accuracies [1].

Instance reduction methods can be categorized as retaining either internal or border instances:

- Border instances: the intuition for retaining border instances is that internal instances do not affect the decision boundaries and thus can be removed with relatively little effect on classification.

- Internal instances: the intuition for retaining internal instances is that removing border instances will hopefully removes instances that are noisy.

In our experiments, we focus on three reduction algorithms that performed well in reducing the number of instances [2], and provided good results before applying Neural Network learning [19]. These algorithms eliminate border instances which tend to be noisy instances or hard to learn untypical instances.

A. *The Edited nearest neighbor algorithm*

Edited Nearest Neighbor ENN [20] is a decremental algorithm which removes an instance if it does not agree with the majority of its k nearest neighbors (with k= 3). This removes noisy instances as well as near border instances and retains all internal instances. Figure 1 shows the pseudo code for ENN algorithm.

B. *AllKnn*

AllKnn [2] is a batch algorithm which makes k iterations, at the ith iteration; it flags as bad any instance that is not classified correctly by its i nearest neighbors. After completing all iterations, the algorithm removes all instances flagged as bad. Figure 2 shows the pseudo code for AllKnn algorithm.

```
For each instance(i)
    If(the class of instance(i) <> the majority class of k nearest neighbor)
      Remove the instance
```

Fig. 1. Pseudo-code for ENN algorithm.

```
oldk = k
For each instance (i)
  For k=1 till oldk
     If (the class of instance (i) <> the majority class of k nearest neighbors)
         Flag the instance for pruning
Remove each flagged instance
```

Fig. 2. Pseudo-code for AllKnn algorithm.

```
Let T be the initial set of instances
Measure the distance of each instance in T from its nearest enemy (instance
with different class). Sort the instances in T by their distance, in ascending
order.
Let S = T.
 For each instance P in S:
      Find P.N1..k+1, the k+1 nearest neighbors of P in S.
      Add P to each of its neighbors' lists of associates.
 For each instance P in S:
      Let with= # of associates of P classified correctly with P as a neighbor.
      Let without= # of associates of P classified correctly without P.
      If without >= with
            Remove P from S.
            For each associate A of P
                 Remove P from A's list of nearest neighbors.
                 Find a new nearest neighbor for A.
                 Add A to its new neighbor's list of associates.
 Return S.
```

Fig.3. Pseudo-code for DROP5 algorithm.

C. *DROP5*

DROP5 [1] is a decremental algorithm which removes the instance "S" if at least as many of its associates (i.e., instances which have "S" on their nearest neighbor list) are classified correctly without it. It considers removing first the instances that are nearest to their nearest enemy (i.e., instance from different class), and proceeding outward. By removing points near the decision boundary first, the decision boundary is smoothed. Figure 3 shows the pseudo code for DROP5 algorithm.

IV. EMPIRICAL RESULTS FOR RULE INDUCTION METHODS USING THE REDUCED SET

Our objective is to apply some instance reduction methods before applying the different rule induction algorithms and compare the results with and without applying the reduction.

A. *Methods*

We applied the three methods for instance reduction (AllKnn, ENN and DROP5) that are intended to remove the border and noisy instances before using the CN2, PRISM and RISE. We also apply DROP5 [1] method on instances flagged by AllKnn to be removed and we call this method as AllKnnDROP5 method.

To test if these methods will affect the accuracy of the CN2, PRISM and RISE algorithms, we conducted experiment on a collection of Machine Learning data sets available from the repository at University of California at Irvine [21]. Predictive accuracy was estimated using 10-fold cross-validation [22] and we used the same folds for each rule induction method. Instance-removal was performed separately for each fold of the cross-validation. Results were compared using statistical paired t-test with confidence 0.05. For each pre prune method, we counted the number of datasets where the predictive accuracy has been statistically improved (win) or statistically reduced (loss).

B. *Results*

We investigate the effect of preceding instance reduction methods on the complexity of rule set (roughly represented here by the number of generated rules). Figure 4 shows that for all rule induction methods, the number of generated rule has been reduced after applying different instance reduction methods. It is clear that applying DROP5 achieved the largest reduction in number of generated rules.

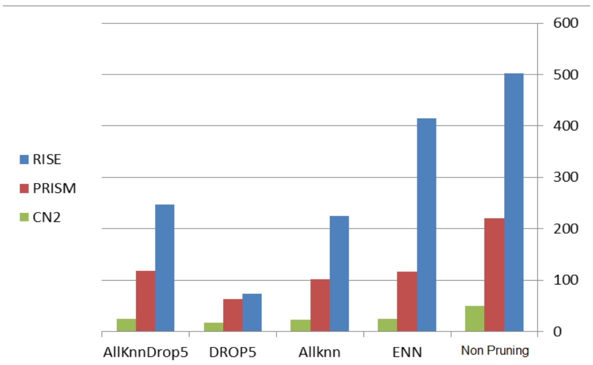

Fig. 4. Comparing the average number of generated rules before and after applying instance reduction methods for different rule induction.

Table II shows the results obtained for CN2 and applying the four prepruning methods with respect to the predictive accuracy. Our experiments show that there is no statistically significant effect on predictive accuracy after applying ENN, AllKnn and AllKnnDrop5 on 19, 19 and 20 datasets respectively. There is a statistically significant increase in predictive accuracy for 2 datasets. We can conclude that preceding CN2 with these instance reduction methods does not adversely affect the predictive accuracy on most datasets and, for two datasets, it enhances the predictive accuracy. However, when using DROP5, there is no statistically significant increase in predictive accuracy for any of the datasets. Furthermore, for 15 of the 22 datasets, using DROP5 leads to statistically significant decrease.

Table III summarizes the effect of instance selection (pruning training data) on the generalization of RISE algorithm. Our experiments show that the predictive accuracy is not statistically affected after applying ENN, AllKnn, DROP5 and AllKnnDrop5 on 17, 16, 8 and 17 datasets respectively. Applying ENN, AllKnn, and AllKnnDrop5 gave statistically significant increases in predictive accuracy on 3, 4 and 3 datasets respectively. But applying DROP5 produced the worst results and it is not recommended as pre pruning method for RISE rule induction.

Table IV clearly shows that applying ENN, AllKnn, DROP5 and AllKnnDrop5 before PRISM [23] does not statistically affect the predictive accuracy on 11, 14, 9 and 15 datasets respectively. The results reveal that applying ENN, AllKnn and AllKnnDrop5 gave statistically significant increase on 9, 7 and 6 datasets respectively. Applying DROP5 produced the worst results and it is not recommended to be used as pre pruning method for PRISM rule induction.

TABLE II: EMPIRICAL RESULTS COMPARING PREDICTIVE ACCURACY USING ALLKNN ENN, DROP5 AND ALLKNNDROP5 PREPRUNING WITH CN2.

Data Sets	Without pruning	ENN	AllKnn	DROP5	AllKnnDrop5
Iris	89.98	92.00	92.67	80.67	93.34
Voting	95.34	95.10	95.33	85.35	95.57
Vowels	67.11	65.97	66.75	85.07	67.31
Heart Cleveland	80.66	76.66	77.33	71.66	79.34
Glass	64.76	58.05	61.98	51.92	66.22
Liver disorders	66.77	64.11	65.64	60.3	66.52
Wine	91.77	94.11	93.52	70	95.28
Pima Indians Diabetes	70.3	73.16	74.7	73.4	72.1
Promoters	85.00	81.00	80.00	63	80.00
Hepatitis	78.65	80.00	80.00	52.67	79.34
Vehicle	57.85	60.10	60.71	54.99	60.10
pole-and-cart	61.68	63.88	66.24	62.56	63.51
Blood Transfusion Service	75.68	76.61	76.35	73.11	75.96
Ecoli	79.10	83.31	80.91	73.34	80.90
Soybean	86.32	82.67	83.01	63	83.32
ZOO	92.00	87.00	90.00	81	89.00
Yeast	48.98	55.47	56.43	51.82	56.56
Led Creator	72.30	72.30	71.30	68.9	71.90
vertebral_column	80.96	83.21	81.28	81.28	82.24
Ionosphere	89.43	85.71	86.56	53.71	85.71
Wave	69.70	70.38	70.74	67.96	71.38
Balance Scale	75.30	74.70	74.34	67.1	74.34
Average	76.35	76.16	76.63	67.86	76.82
Win/tie/loss		2/19/1	2/19/1	0/7/15	2/20/0

V. RELATED WORK

Using the noise filtering methods to reduce the border instances before applying the induction method can remove the noisy instances and smooth the decision boundaries. This may improve the predictive accuracy for the induction method. El Hindi and Alakhras [19] showed that filtering out border instances before training an artificial neural network will improve the predictive accuracy and speed up the training process by reducing the training epochs. Gamberger et al. investigated the effect of a new noisy instance detection method before induction on a specific dataset (i.e., early diagnosis of rheumatic diseases) [24]; this method is suitable for datasets with just two classes. Grudzinski concentrated on the EkP system [25] as an instance reduction method before rule induction, and they illustrated it is possible to extract simpler sets of rules from reduced datasets.

VI. CONCLUSION

In this paper, we extended our previous work [26] by investigating preceding three different types of rule induction with instances reduction methods. The search strategies used by the three algorithms vary in terms of both type (depth-first or beam search) and direction (general-to-specific or specific-to-general). Our results show that applying instance reduction techniques as a pre-pruning process for rule induction will reduce the number of generated rules without adversely affecting the predictive accuracy and may improve it in some cases. For future work, we recommend investigating whether it would be beneficial to use other instance reduction methods that conduct instance pruning more carefully such as c-pruner [27]. We also highly recommend investigating the effect of preceding the instance reduction methods with rule induction on noisy datasets.

TABLE III: EMPIRICAL RESULTS COMPARING PREDICTIVE ACCURACY USING ALLKNN ENN, DROP5 AND ALLKNNDROP5 PREPRUNING WITH RISE.

Data Sets	Without pruning	ENN	AllKnn	DROP5	AllKnnDrop5
Iris	95.33	94.00	94.67	94.01	94.67
Voting	95.10	95.32	95.79	93.25	95.32
Vowels	92.68	88.87	89.25	85.97	89.63
Heart Cleveland	77.00	77.01	75.32	71.01	75.01
Glass	67.14	62.85	64.77	52.37	65.70
Liver disorders	65.29	61.18	62.00	57.05	65.23
Wine	97.64	95.28	96.46	88.83	97.64
Pima Indians Diabetes	67.63	68.29	68.37	68.56	67.70
Promoters	86.00	92.00	88.00	67.00	87.00
Hepatitis	80.67	80.67	80.66	52.00	80.67
Vehicle	70.35	68.47	66.55	65.36	67.62
pole-and-cart	61.87	82.18	65.49	58.81	64.24
Blood Transfusion Service	73.92	79.19	77.84	74.87	77.34
Ecoli	84.76	85.75	85.46	83.02	86.35
Soybean	91.00	87.67	87.66	82.67	88.33
ZOO	96.00	89.00	93.00	89.00	93.00
Yeast	52.97	57.56	58.25	53.99	56.83
Led Creator	72.60	72.40	72.60	69.40	72.80
vertebral_column	82.91	81.60	81.93	81.30	82.90
Ionosphere	92.56	91.42	91.71	77.42	90.56
Wave	81.84	82.18	83.26	79.06	82.82
Balance Scale	78.06	81.13	80.97	77.75	81.62
Average	80.15	80.64	80	73.76	80.14
Win/tie/loss		3/17/2	4/16/2	0/8/14	3/17/2

TABLE IV: EMPIRICAL RESULTS COMPARING PREDICTIVE ACCURACY USING ALLKNN ENN, DROP5 AND ALLKNNDROP5 PREPRUNING WITH PRISM.

Data Sets	Without pruning	ENN	AllKnn	DROP5	AllKnnDrop5
Iris	91.40	88.20	88.80	79.20	88.80
Voting	92.50	95.50	95.70	93.10	96.20
Vowels	52.40	50.70	51.10	42.40	51.10
Heart Cleveland	68.00	74.00	73.90	62.70	72.40
Glass	43.90	47.20	48.70	32.90	48.30
Liver disorders	47.90	56.90	53.60	51.20	52.40
Wine	86.30	83.90	83.90	69.80	86.30
Pima Indians Diabetes	62.80	63.20	64.00	60.40	63.40
Promoters	73.00	77.00	74.00	52.00	72.00
Hepatitis	69.30	78.70	77.30	79.30	74.60
Vehicle	58.70	57.60	59.30	50.00	59.30
pole-and-cart	52. 50	56.20	56.60	48.70	55.00
Blood Transfusion Service	71.70	76.4	72.70	69.20	73.20
Ecoli	73.30	79.00	78.40	69.60	78.40
Soybean	79.50	73.90	73.40	56.30	74.20
ZOO	92.00	84.00	88.00	85.00	87.00
Yeast	43.80	49.30	46.40	41.70	46.70
Led Creator	71.70	72.40	71.60	67.40	72.10
vertebral_column	73.40	78.00	74.20	75.40	75.50
Ionosphere	86.90	87.50	89.30	53.30	88.80
Wave	59.30	63.10	63.10	54.30	63.50
Balance Scale	62.70	72.10	73.00	52.30	73.00
Average	69.55	71.13	70.77	61.19	70.55
Win/tie/loss		9/11/2	7/14/1	1/9/12	6/15/1

REFERENCES

[1] D. Wilsson and T. Martinez, "Reduction techniques for instance based learning algorithms". *Machine Learning*, vol. 38 no.3, pp. 257-286, 2000.

[2] D. Wilsson and T. Martinez, "Instance Pruning Technique". *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, ed. Douglas H. Fisher, pp. 403-411. Morgan Kauffmann. San Francisco, CA, 1997.

[3] D. Pham, "A novel rule induction algorithm with improved handling of continuous valued attributes". Ph.D. dissertation, Cardiff Univ., Cardiff, 2012.

[4] R. Michalski and K. Kaufman, "The AQ19 system for machine learning and pattern discovery: A general description and user guide". *Reports of the Machine Learning and Inference Laboratory, MLI 01-2*, George Mason University, Fairfax, VA, USA, 2001.

[5] P. Clark and R. Boswell. "Rule induction with CN2: some recent improvements". In: Kodratoffy (Ed.). Lecture Notes in Computer Science (LNCS). *Proceedings of the sixth European working session on learning*, vol. 482, pp. 151-163 Springer-Verlag, Portugal, 1991,.

[6] P. Clark and T. Niblett. "The CN2 induction algorithm", *Machine Learning*, vol. 3, pp. 261-283, 1989.

[7] W. Cohen, "Fast effective rule induction". *Machine Learning: Proceedings of the 12th international conference*, eds. Armand Prieditis, Stuart J. Russell, pp. 115-123. Lake Tahoe, California: Morgan Kaufmann, 1995.

[8] K. Shehzad, "New Rule Induction Algorithms with Improved Noise Toleranceand Scalability". Ph.D. dissertation, Systems Engineering Division, University of Wales Cardiff, Cardiff, UK. 2009.

[9] P. Domingos, "The RISE system: conquering without separating". *Proceeding of the sixth IEEE International Conference on tools with artificial intelligence*, pp. 704-707. New Orleans, LA: IEEE computer society Press, 1994.

[10] D. W. Aha, D. Kibler and M. K. Albert. "Instance – based learning algorithm", Machine Learning, vol. 6, pp. 37-66, 1991.

[11] S. Weiss and N. Indurkhya. "Reduced complexity rule induction". *In Proceedings of 12th International joint conference on Artificial Intelligence*, eds. John Mylopouslos and Ray Reiter. pp. 678-684. Morgan Kauffmann. Sydney, Australia, 1991.

[12] J. Fürnkranz and G. Widmer."Incremental reduced error pruning". *Proceedings of the 11th International Conference on Machine learning*, eds. Cohen W. and Hirsh H., pp.70–77. Morgan Kaufmann. New Brunswick, NJ, 1994.

[13] J. Cendrowska "PRISM: An Algorithm for Inducing Modular Rules". International Journal of Man-Machine Studies, eds. Motta E., Wiedenbeck S. vol. 27(4), pp. 349-370, 1987.

[14] J. Quinlan, "Discovering rules by induction from large collections of examples". *Expert systems in the micro-electronics age*, ed. Michie, Edinburgh University press. pp. 168-201, 1979.

[15] P. E Hart. "The condensed nearest neighbor rules". Institute of Electrical and Electronics Engineers Transactions on Information theory IEEE, vol. 14 no.3, pp. 515-516, 1968.

[16] G. L. Ritter, H. B. Woodruff, S. R. Lowry and T. L. Isenhour. "An Algorithm for a Selective Nearest Neighbor Decision Rule". *IEEE Transactions on Information Theory*, vol. 21, no. 6: pp. 665-669, 1975.

[17] G. W. Gates, "The reduced nearest neighbor rule". In*stitute of Electrical and Electronics Engineers Transactions on Information theory IEEE*, vol. 18 no.3, pp. 431-433, 1972.

[18] I.. Tomek, "An experiment with the edited nearest-neighbor rule". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6 no. 6: pp. 448-452, 1976.

[19] K. El Hindi and M. Al-Akhras "Eliminating border instance to avoid overfitting", eds. Antonio P Alakhras, Alma dos Reis. *Proceeding of Intelligent Systems and Agents*. pp 93 – 99. IADIS press Algarve, Portugal, 2009.

[20] D. L. Wilson, "Asymptotic properties of nearest neighbor rules Using Edited Data". I*EEE Transactions on Systems, Man, and Cybernetics*, vol. 2 no.3: pp. 408-421, 1972.

[21] P. M. Murphy and D. W. Aha. "UCI repository of Machine Learning Data bases", 1994.available by anonymous ftp to ics.uci.edu in the pub/machine-learning-databases directory.

[22] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection". *In proceedings of 14th international joint conference on artificial intelligence*, ed. Mellish C. pp. 1137-1143. Morgan Kaufmann. San Francisco, CA, USA, 1995.

[23] M. A. Bramer, "Inducer: a rule induction workbench for data mining", *Proceedings of the 16th IFIP World Computer Congress Conf. Intelligent Information Processing*, eds. Z. Shi, B. Faltings and M. Musen, pp. 499-506, Publishing House of Electronics Industry (Beijing),2000,.

[24] D. Gamberger, N. Lavrac and S. Dzeroski. "Noise Elimination in inductive concept learning: A case study in medical diagnosis". Lecture Notes in Computer Science (LNCS). *7th International Workshop, ALT '96 Sydney*, vol. 1160, pp. 199-212: Springer-Verlag, Berlin, 1996.

[25] K. Grudzinski, "EkP: A fast minimization – based prototype selection algorithm". *In Intelligent Information System XVI*, pp. 45-53. Academic Publishing House EXIT. Warsaw, 2008.

[26] O. Othman and C. Bryant. "Preceding Rule Induction with Instance Reduction Methods", *Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, ed. Perner, P, Springer-Verlag, Berlin, Germany, pp.209-218, New York, USA, 2013.

[27] K. Zhao, S Zhou, J. Guan and A. Zhou, "C-Pruner: An improved instance pruning algorithm". *Proceedings of the 2th International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 94–99, 2003, Sheraton Hotel, Xi'an, China: Piscataway, NJ: IEEE, 2003.

Christopher H. Bryant received a BSc (Hons) in Combined Studies in Science from Sunderland Polytechnic, UK (1990), an MSc in Applied Artificial Intelligence from the University of Aberdeen, UK (1993) and a PhD from the University of Manchester Institute of Science and Technology, UK (1996). He is a lecturer at the University of Salford, UK. Previously, he was a lecturer at the Robert Gordon University, Aberdeen and a research associate at the University of York. His research interests include the development and application of Machine Learning algorithms. He has co-authored 28 papers, including one in Nature.

Osama M. Othman received a BSc in Computer Science from Jordan University, Jordan (1998), an MSc in Applied Artificial Intelligence from the University of Jordan, Jordan (2001). Currently studying Phd in Artificial Intelligence at University of Salford, UK. He is a branch manager at the Galaxy International Group, Jordan. His research interests include the development and application of Machine Learning algorithms.